



- HOME
- ABOUT
- LOGIN
- REGISTER
- SEARCH
- CURRENT
- ARCHIVES
- ANNOUNCEMENTS

[Home](#) > [Vol 16, No 4](#) > [Prasetyo](#)

## Multi-class K-support Vector Nearest Neighbor for Mango Leaf Classification

*Eko Prasetyo, R. Dimas Adityo, Nanik Suciati, Chastine Fatchah*

### Abstract

K-Support Vector Nearest Neighbor (K-SVNN) is one of methods for training data reduction that works only for binary class. This method uses Left Value (LV) and Right Value (RV) to calculate Significant Degree (SD) property. This research aims to modify the K-SVNN for multi-class training data reduction problem by using entropy for calculating SD property. Entropy can measure the impurity of data class distribution, so the selection of the SD can be conducted based on the high entropy. In order to measure performance of the modified K-SVNN in mango leaf classification, experiment is conducted by using multi-class Support Vector Machine (SVM) method on training data with and without reduction. The experiment is performed on 300 mango leaf images, each image represented by 260 features consisting of 256 Weighted Rotation- and Scale-invariant Local Binary Pattern features with average weights (WRSI-LBP-avg) texture features, 2 color features, and 2 shape features. The experiment results show that the highest accuracy for data with and without reduction are 71.33% and 71.00% respectively. It is concluded that K-SVNN can be used to reduce data in multi-class classification problem while preserve the accuracy. In addition, performance of the modified K-SVNN is also compared with two other methods of multi-class data reduction, i.e. Condensed Nearest Neighbor Rule (CNN) and Template Reduction KNN (TRKNN). The performance comparison shows that the modified K-SVNN achieves better accuracy.

### Keywords

support vector machine, k-support vector nearest neighbor, data reduction, mango leaves, entropy

### Full Text:

[PDF](#)

DOI: <http://dx.doi.org/10.12928/telkomnika.v16i4.8482>

### Article Metrics

Abstract view : 186 times  
PDF - 117 times

### Refbacks

There are currently no refbacks.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

**TELKOMNIKA Telecommunication, Computing, Electronics and Control**  
 ISSN: 1693-6930, e-ISSN: 2302-9293  
 Universitas Ahmad Dahlan, 4th Campus, 9th Floor, LPPI Room  
 Jl. Ringroad Selatan, Kragilan, Tamanan, Banguntapan, Bantul, Yogyakarta, Indonesia 55191  
 Phone: +62 (274) 563515, 511830, 379418, 371120 ext. 4902, Fax: +62 274 564604

**02122632**

[View TELKOMNIKA Stats](#)

**USER**

Username:

Password:

Remember me

SJR 2018 : 0.283 (Q2)  
 CiteScore 2018 : 1.09  
 SNIP 2018 : 0.730

TELKOMNIKA is the best journal in Indonesia 2017

**Telkomnika**

Q2 Electrical and Electronic Engineering  
 best quartile

SJR 2018  
 0.28

powered by scimagojr.com

**TEMPLATE**

JOURNAL TEMPLATE

- QUICK LINKS**
- Author Guideline
  - Editorial Boards
  - Reviewers
  - Online Submissions
  - Abstracting and Indexing
  - Scopus: Add missing document
  - Publication Ethics
  - Visitor Statistics
  - Contact Us

**JOURNAL HARDCOPY**

Order journal prints (hardcopy)

<<click in here>>

**ICW-TELKOMNIKA**

**2020 ICW-TELKOMNIKA**  
 INTERNATIONAL CONFERENCE

**JOURNAL CONTENT**

Search:

Search Scope:

Browse

- By Issue
- By Author
- By Title
- Other Journals

## Multi-class K-support Vector Nearest Neighbor for Mango Leaf Classification

Eko Prasetyo<sup>\*1</sup>, R. Dimas Adityo<sup>2</sup>, Nanik Suciati<sup>3</sup>, Chastine Fatichah<sup>4</sup>

<sup>1,2</sup>Department of Informatics, Engineering Faculty, University of Bhayangkara Surabaya  
Jl. Ahmad Yani 114, Surabaya 60231, Indonesia

<sup>3,4</sup>Department of Informatics, Faculty of Information Technology and Communication  
Institut Teknologi Sepuluh Nopember, Kampus ITS, Jl. Raya ITS, Sukolilo, Surabaya 60111, Indonesia

\*Corresponding author, e-mail: eko@ubhara.ac.id<sup>1</sup>, dimas@ubhara.ac.id<sup>2</sup>, nanik@if.its.ac.id<sup>3</sup>,  
chastine@if.its.ac.id<sup>4</sup>

### Abstract

*K-Support Vector Nearest Neighbor (K-SVNN) is one of methods for training data reduction that works only for binary class. This method uses Left Value (LV) and Right Value (RV) to calculate Significant Degree (SD) property. This research aims to modify the K-SVNN for multi-class training data reduction problem by using entropy for calculating SD property. Entropy can measure the impurity of data class distribution, so the selection of the SD can be conducted based on the high entropy. In order to measure performance of the modified K-SVNN in mango leaf classification, experiment is conducted by using multi-class Support Vector Machine (SVM) method on training data with and without reduction. The experiment is performed on 300 mango leaf images, each image represented by 260 features consisting of 256 Weighted Rotation- and Scale-invariant Local Binary Pattern features with average weights (WRSI-LBP-avg) texture features, 2 color features, and 2 shape features. The experiment results show that the highest accuracy for data with and without reduction are 71.33% and 71.00% respectively. It is concluded that K-SVNN can be used to reduce data in multi-class classification problem while preserve the accuracy. In addition, performance of the modified K-SVNN is also compared with two other methods of multi-class data reduction, i.e. Condensed Nearest Neighbor Rule (CNN) and Template Reduction KNN (TRKNN). The performance comparison shows that the modified K-SVNN achieves better accuracy.*

**Keywords:** Support vector machine, K-Support vector nearest neighbor, Data reduction, Mango leaves, Entropy

Copyright © 2018 Universitas Ahmad Dahlan. All rights reserved.

### 1. Introduction

The previous research in mango leaf classification used image texture as the feature with K-Nearest Neighbour (K-NN) and Artificial Neural Network (ANN) Back-propagation as the classification methods [1]. Furthermore, another research in [2] tried to improve the performance by selecting some important texture features using Fisher's Discriminant Ratio, and K-NN as the classification method. In this research, we conduct improvement by combining texture, colour, and shape features, in order to involve more rich information contained in leaf image. For the texture, colour, and shape features, we use Weighted Rotation- and Scale-invariant Local Binary Pattern with average weight (WRSI-LBP-avg) [3]; average and standard deviation [4] of gray-scale intensities; compactness [5,6] and Circularity [4] of the leaf image; respectively. Combining those three low-level features increase the feature size, so that an effort to reduce the data in order to speed up the computation is required.

Data reduction can be conducted by using several methods, such as Condensed Nearest Neighbour Rule (CNN) [7], Template Reduction KNN (TRKNN) [8], and K-Support Vector Nearest Neighbour (K-SVNN) [9]. The CNN algorithm finds a subset of training data, in which the distance of each member in the same class has shorter distance compared to members in different classes. The TRKNN [8] method uses concept of nearest neighbours chains to reduce the training data. While the K-SVNN solves the problem by removing data that has no impact to the decision boundary [9] or data with low Significant Degree (SD). Comparing with other methods [10-12], the K-SVNN is good enough for data reduction while preserve the accuracy. Generally, applying data reduction process on data training can speed up the training process. Unfortunately, K-SVNN method works only for binary class. To solve the multi-class

problem, the authors propose entropy to calculate the SD. Entropy can measure the impurity of data class distribution, so that the selection of the SD can be conducted based on the high entropy. The data with same class distribution has zero impurity, whereas data with uniform class distribution has the highest impurity [13]. We use impurity value presented by Entropy to calculate SD property of each data. The entropy concept can provide an estimation to the SD values. A data with the lowest SD value, which is equal to zero, is a data with no impact to decision boundary. Thus, all data having SD or entropy equal to zero, can be removed from the list of training data. The remaining data will become the result of the data reduction process.

In this research we classify 3 mango varieties, namely Gadung, Jiwo, and Manalagi, so that we need methods that can work for multi-class classification, such as K-NN, and Support Vector Machine (SVM) [13-15]. SVM is a convex optimization problem, which is an efficient algorithm to find the minimum global objective function. SVM also performs capacity control by maximizing the margin of decision boundary [13]. Apart from all that, the initial SVM design is for binary classes. In order to apply SVM to multi-class problems, we use some binary SVM with a multi-class solution design scheme. Examples of such schemes are one-against-all, one-against-one, and error correcting output code [13, 16]. To measure classification performance on data with and without reduction, an experiment is performed on 300 mango leaf images, each image represented by 260 features consisting of 256 Weighted Rotation- and Scale-invariant Local Binary Pattern features with average weights (WRSI-LBP-avg) texture features, 2 color features, and 2 shape features. The authors also conduct testing on Android applications using an emulator software and a mobile phone with camera to classify mango leaf. In addition, performance comparisons with two other data reduction methods, namely CNN [7] and TRKNN [8], are also carried out.

## 2. Literature Review of Data Reduction

### 2.1. K-support Vector Nearest Neighbor

K-Support Vector Nearest Neighbor (K-SVNN) was introduced in the research [9] to produce support vectors. The support vectors have different concept compared to the support vector in SVM. These support vectors are part of training data that have an impact to the hyperplane function. To get the support vectors, K-SVNN uses two properties, they are score and significant degree (SD). The score property is divided into two parts, Left Values (LV) for the same class, and Right Value (RV) for different classes of neighbors. The initial value for LV and RV is zero. This value will be added by 1 when each data is selected as the nearest neighbor of the other data. If the class of two data is equal then LV is added by 1, else then RV is added by 1. If  $C$  is the set of data classes,  $c_i$  is the class of data  $x_i$ ,  $c_j$  is the class of data  $x_j$ , then suppose  $x_i$  is selected as one of the nearest neighbors  $x_j$ . The value of  $LV_i$  and  $RV_i$  of data  $x_i$  is added by 1 refers to equation 1.

$$LV_i = LV_i + \text{diff}(LV, c_i, c_j) \quad (1)$$

$$RV_i = RV_i + \text{diff}(RV, c_i, c_j)$$

Where  $\text{diff}(LV, c_i, c_j)$  has value 1 if the class of data  $x_i, c_i$  is equal to the class of data  $x_j, c_j$ . Conversely, it has value 0 if the class is different. While  $\text{diff}(RV, c_i, c_j)$  is 0 if the class of data  $x_i, c_i$  is equal to the class of data  $x_j, c_j$ . And vice versa, the value is 1 if the class of two data is different. As presented in the equation 2.

$$\text{diff}(LV, c_i, c_j) = \begin{cases} 1 & , \text{if } c_i = c_j \\ 0 & , \text{if } c_i \neq c_j \end{cases}$$

$$\text{diff}(RV, c_i, c_j) = \begin{cases} 0 & , \text{if } c_i = c_j \\ 1 & , \text{if } c_i \neq c_j \end{cases} \quad (2)$$

The SD property relates to the relevance level of the training data to the decision boundary. The value range is 0 to 1. As the higher value the more relevant the train data to the decision boundary. This research uses the threshold  $T > 0$ , this means that any small value of relevance will be used as support vector. The value of SD is obtained from the equation 3.

$$SD_i = \begin{cases} 0 & , LV_i = RV_i = 0 \\ \frac{LV_i}{RV_i} & , LV_i < RV_i \\ \frac{RV_i}{LV_i} & , LV_i > RV_i \\ 1 & , LV_i = RV_i \end{cases} \quad (3)$$

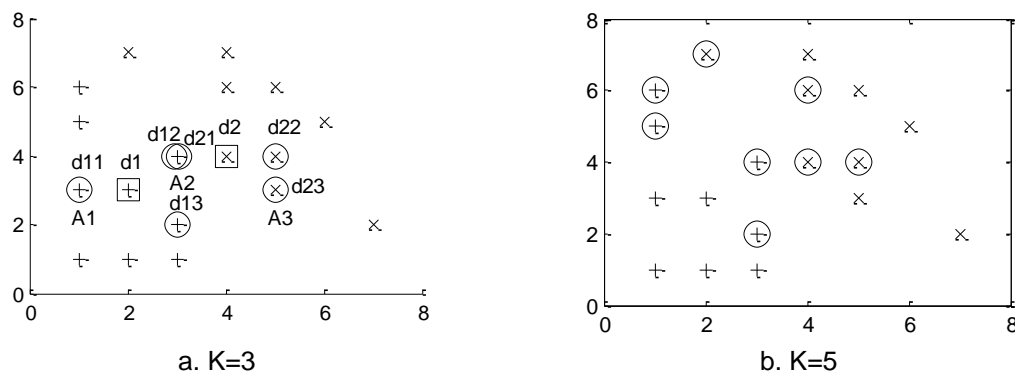


Figure 1. K-SVNN [9]

An example process of obtaining the property score is presented in Figure 1(a). The figure is obtained when K-SVNN uses  $K=3$ . The training data  $d_1$  has 3 nearest neighbors  $d_{11}$ ,  $d_{12}$ ,  $d_{13}$ , since the class of 3 neighbors is equal to  $d_1$  (class +), the LV values for each data neighbors  $d_{11}$ ,  $d_{12}$  and  $d_{13}$  added 1, whereas the RV does not, while the training data  $d_2$  has the nearest neighbor  $d_{21}$ ,  $d_{22}$ ,  $d_{23}$ . For the data neighbors  $d_{22}$  and  $d_{23}$  the class is equal to  $d_2$  (class x) then LV value in the neighbor data  $d_{22}$  and  $d_{23}$  added 1, whereas the neighbor data  $d_{21}$  the RV value increases 1. The result of the train data  $d_1$  and  $d_2$  only, for data A1 value  $LV=1$  and  $RV=0$ , for data A2 values  $LV=1$  and  $RV=1$ , while data A3 values  $LV=1$  and  $RV=0$ . The process is performed on all training data [9].

Like the K-NN generally, K-SVNN is also influenced by the K as the basis of nearest neighbor. The value suggested by previous research [9] is  $K > 1$ . As K gets bigger then the data released from the training data are also less, and vice versa. The smaller reduction results also do not guarantee higher accuracy. So the determination of the K also needs a high consideration.

The K-SVNN training algorithm can be explained as follows [9]:

1. Initialization: the D is set of training data, the K is a number of nearest neighbors, the T is selected SD threshold, LV and RV set to 0 respectively for all training data.
2. For every training data  $d_i \in D$ , do step 3 until 5.
3. Calculate dissimilarity (distance) from the  $d_i$  to other training data.
4. Select  $dt$  as the nearest neighbour of train data (excluded  $d_i$ ).
5. For each train data in  $dt$ , if the class label is equal to  $d_i$ , then add 1 to  $LV_i$ , else then add 1 to  $RV_i$ . use equation 1.
6. For each train data  $d_i$ , calculate  $SD_i$  using equation 3.
7. Select train data with  $SD \geq T$ , save in memory (variable) as a template for prediction.

The results of the support vector search testings are presented in Figure 1(b). This search is performed using  $K=5$ , with Euclidean distance. Support vectors obtained are indicated by the circle symbols (o).

The K-SVNN has been studied where the result of data reduction can help the performance of classification methods. The research is conducted performance comparison the accuracy and time spent for training and prediction [10]. The other methods compared are Decision Tree and Naive Bayes. The accuracy obtained by classifier with data reduction result is better than the others, but the time spent is longer. The performance comparison to SVM and Artificial Neural Network Error Back-Propagation (ANN-EBP) show the data reduction result used by K-NN is relatively higher accuracy compared to the others [11].

## 2.2. Other Data Reduction Method

Generally, the data should go through the pre-processing to prepare the data in order to support optimal classification results [16]. The most commonly pre-processing are normalization, binary, discretization, sampling, faulty data settlement, dimensional reduction, and data reduction. The example research for feature is Correlation-based Feature Selection (CFS) method for selecting shape features, colors and textures [17]. The data reduction as pre-processing is also important work. The use of K-SVNN as part of pre-processing of training data before use during training on ANN-EBP was also done [18]. The results show that K-SVNN can be used as data pre-processing to reduce the training data while preserve training results quality. The use of K-SVNN as pre-processing indicates that training time is reduced 15% to 80%, whereas the prediction accuracy decreased 0% to 4.76%.

The data reduction research related to Condensed Nearest Neighbor Rule (CNN) algorithm [7]. The goal is to get some data that has an impact to the classification decisions. The algorithm begins by forming the initialization of the reduction results  $S$  by selecting one data from each class from the training data  $TR$ . Furthermore, each training data  $x_i$  from  $TR$  is classified using the reduction data  $S$ . If the class is not same as the training data  $x_i$  then the training data is added to  $S$ . The step is iteratively repeated on all training data until no misclassified training data. The CNN algorithm gets many updates to improve its work quality, such as Tomek Condensed Nearest Neighbor (TCNN) [19] and Template Reduction KNN (TRKNN) [8]. In TCNN, the training data  $x_i$  that gets wrong classification because  $s_j$  (of  $S$ ) is selected, then it isn't added the  $x_i$  training data into  $S$ , instead of look for the same class nearest neighbour of  $s_j$  in the  $TR$  and add it to  $S$  [19]. The TRKNN method introduces the concept of nearest neighbour chains. Every chain is related to one instance, and the  $S$  is built by searching the nearest neighbors of each element of the chain, which belongs alternatively to the class of the starting instance, or to a different class. The chain searching is stopped when an element is selected again to be subset of the chain [8].

## 3. Research Method

The research methodology for mango leaf classification is presented in Figure 2. This research is divided into several stages. They are as follow:

- (1) Image acquisition  
In the first stage, image acquisition is conducted by phone cell camera with resolution 2592x1944. The number of mango leaf in each image is one leaf with normal effect.
- (2) Pre-processing 1  
As the result of the acquisition environment, some of the acquisition results in some parts of the image object exposed to high-intensity light, so we have to remove the area [20].
- (3) Image segmentation  
To get the leaf area, we conduct segmentation by Otsu thresholding method on Cr color component [21].
- (4) Pre-processing 2  
In this stage, we conduct some preprocessing, ie. morphological operations, resizing, cropping, and texture sampling.
- (5) Features extraction  
We generate 3 types of features, ie. texture, color, and shape. For texture features we use WRSI-LBP-avg [3]. For color features use the average and standard deviation of gray area texture intensity. For shape features, we use Compactness and Circularity.

## (6) Data splitting

In this step, we perform data splitting. The data is splitted into 2 parts, ie training data and testing data. We use 50:50 splitting, 50% proportion for training data and 50% proportion for testing data. We also use 2 fold cross validation to calculate accuracy.

## (7) Training Data Reduction

This stage is the topic discussed in this paper. In this study used 300 images of mango leaves, represented by 300 data generated. Each data consists of 260 features, comprising 256 WRSI-LBP-avg for texture features, averages and standard deviations for color features, Compactness and Circularity for shape features. Due to the large number, it is necessary to attempt data reduction to simplify and speed up computation. We explain more detail in the next sub section.

## (8) Training in classifier

In this stage, we conduct training on the system using training data from the results of reduction.

## (9) Prediction

The last stage in our system is the prediction of the testing data.

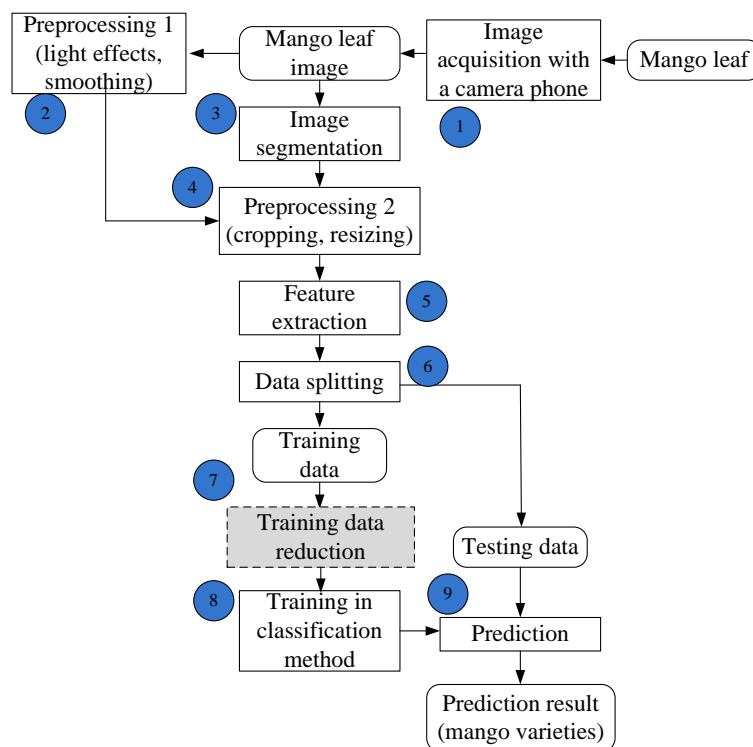


Figure 2. Research Method

### 3.1. Training Data Reduction

As described earlier, this study uses 300 data. Each data is represented by 260 features. Due to large data, it takes effort to reduce the amount of data, especially training data. The reason is because training data would be reference data read during the training process. The amount of large data caused the computation process getting more complex and slow.

The very old data reduction method is Condensed Nearest Neighbor Rule (CNN) algorithm. The goal is to get some data that has an impact to classification decisions. This method has got many modifications to improve the results, such as Tomek Condensed Nearest Neighbor (TCNN) [19] and Template Reduction KNN (TRKNN) [8]. Another simple method based on Nearest Neighbor is K-Support Vector Nearest Neighbor (K-SVNN) [12]. K-SVNN can be used for data reduction based on the selection of Significant Degree (SD). Data with zero SD must be discarded, the rest is the support vector found. Then these data are used as data in

the training process. Unfortunately, this method works only on binary classes. In the case of mango varieties classification, the number of classes is more than two. So K-SVNN modifications are required in order to work on multi-class problem.

### 3.2. Multi-Class K-Support Vector Nearest Neighbor

Generally, the dataset processed in classification is multi-class problem. For example, a dataset with 3 classes, is presented in Figure 3. The figure display 28 data with 2 features, and 3 classes. The classes are represented by symbols plus, diamond, and cross.

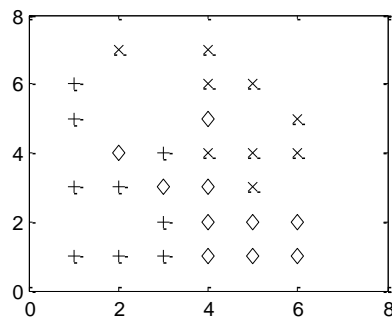


Figure 3. Example of dataset with 3 classes

In multi-class problem of K-SVNN, the authors propose entropy to calculate the SD. Entropy can measure the impurity of data class distribution, so the selection of the SD can be conducted based on the high entropy. The data with same class distribution has zero impurity, whereas data with uniform class distribution has the highest impurity [13]. We need impurity value presented by Entropy to calculate SD property of each data. So, the entropy concept can provide an estimate of the SD values, where the lowest SV value is zero, means it has no impact to decision boundary. Furthermore, for all train data with SD or entropy with zero value are removed from the list of training data, and the rests are the result of the reduction.

The LV and RV scores for each data  $i$  in the multi-class case are replaced by the value of  $V_i(k)$ ,  $k = 1..C$ , where  $C$  is the number of classes. Then the definition of the score for each data  $i$  in class  $k$  is defined by equation 4.

$$V_i(k) = \sum_{j=1}^N I(i, k, j) \quad (4)$$

Where  $I(i, k, j)$  is the result of examination of  $i$ -th data when selected as K nearest neighbor by the  $j$ -th data for class  $k$ . This is done when the  $j$ -th data search for the K nearest neighbor, then the  $i$ -th data is selected. If  $k$  is class label belonging to the  $i$ -th data, same as the  $j$ -th data class then  $V_i(k)$  will be increased 1, but if it is not equal then all classes except  $k$  will be increased 1, as presented in the Equation (5) and (6).

$$I(i, k, j) = \begin{cases} 1, & C_i(k) = C(j) \\ 0, & \text{the others} \end{cases} \quad (5)$$

$$I(i, \sim k, j) = \begin{cases} 1, & C_i(\sim k) = C(j) \\ 0, & \text{the others} \end{cases} \quad (6)$$

Where  $C(j)$  is  $j$ -th data class label that is being processed by the K nearest neighbor.  $C_i(k)$  is  $i$ -th data class label.  $I(i, \sim k, j)$  is the examination result of  $i$ -th data for other class except class  $k$

when selected as K nearest neighbor by the  $j$ -th data. Before calculating the significant degree (SD) value, then V value for each data has to be normalized using the equation 7.

$$V_i^{norm}(k) = \frac{V_i(k)}{\sum_{k=1}^C V_i(k)} \quad (7)$$

The significant degree (SD) value is calculated using entropy of normalized V value, as in equation 8.

$$SD_i = entropy_i = -\sum_{k=1}^C (V_i^{norm}(k) \times \log_2(V_i^{norm}(k))) \quad (8)$$

Furthermore, the support vector is selected from train data with the condition  $SD > T$ . In this research, the authors use  $T=0$ , it means that any SD value would be used as support vector.

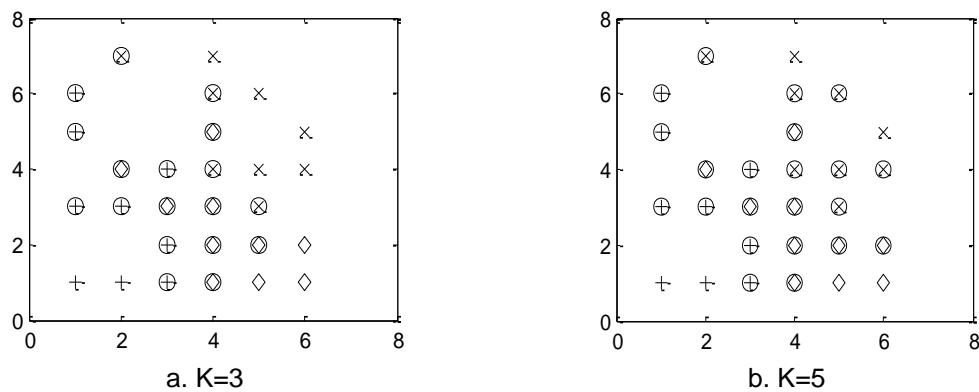


Figure 4. Search result of support vector

The proposed multi-class K-SVNN training algorithm as follows:

1. Initialization: the D is set of train data, the K is number of nearest neighbors, the T is selected SD threshold, the  $V_{ij}$  for all train data in each class is assigned by 0.
2. For each training data  $d_i \in D$ , do step 3 until 5.
3. Calculate dissimilarity (distance) from  $d_i$  to other training data.
4. Select  $dt$  as K nearest neighbour data training (excluded  $d_i$ ).
5. For each train data in  $dt$ , if the class label is equal to  $d_i$ , then add value 1 to  $V_{ij}$  whose class corresponds to the neighbor's train data, otherwise add value 1 to another class  $V_{ij}$ , using equation 4,5 and 6.
6. Do normalization on each  $d_i$  by dividing each class with accumulation of  $d_i$ , using equation 7
7. For each train data  $d_i$ , calculate  $SD_i$  using equation 8
8. Select train data with  $SD \geq T$ , save in memory (variable) as the template for prediction.

The search results support vector with K-SVNN for K=3 and K=5 are presented in Figure 4. When K=3, 18 of 28 data are selected as support vector, indicated by the circle symbol. While the 10 other data are not selected or removed from the data. When K=5, 22 of 28 data are selected.

### 3.3. The Experiments

The authors conducted experiment on 300 mango leaf images, each image represented by 260 features consisting of 256 Weighted Rotation- and Scale-invariant Local Binary Pattern features with average weights (WRSI-LBP-avg) texture features, 2 color features, and 2 shape features. We use 2 fold cross validation to test the performance of multi-class K-SVNN with and



without reduction. We use 50:50 splitting, 50% proportion for training data and 50% proportion for testing data. For classification, we use Support Vector Machine (SVM). The kernel parameters used are Linear, Radial Basis Function (RBF), and Polynomial. All tests performed in this study using data reduction with the choice of K were 3, 6, 9, 12, 15, 18, 21, 24, 27, and 30.

We also measure the reduction rate obtained and time spent in each the K options used to know how much the reduction rate obtained. The wqwAnd to know the performance of new modifications to K-SVNN, we also compare K-SVNN versus other data reduction methods, namely CNN and TRKNN. We also use 2 fold cross validation. The time spent by K-SVNN is required to know the amount of time it should be provided during the reduction process because the data reduction process becomes an additional process in the classification stage.

## 4. Results and Discussion

### 4.1. Data Reduction Using Multi-class K-SVNN

The author performs reduction rate testing by calculate the percentage of data released from the original training data. The results of reduction rate testing and reduction time are presented in Table 1. As the result, the shortest time used by multi-class K-SVNN in data reduction is 54.81 ms for K=3, while the longest time required K-SVNN was 209.13 ms for K=30. The K-SVNN reduction rate was significant, where the highest reduction was 62.00% for K=3 and moved down to 12.00% for K=30. The reduction rate is inversely proportional to the K, this is because a higher number of neighbors involved into more nearest neighbors data that passes into support vector (less percentage reduction). These results can be concluded that data reduction with multi-class K-SVNN can reduce data significantly.

Table 1. Time Comparison (Mili Second), Reduction (Percentage)

K	K-SVNN	
	Reduction (%)	Reduction time (ms)
3	<b>62.00</b>	<b>54.81</b>
6	42.00	64.57
9	30.00	76.48
12	20.67	86.70
15	20.67	117.89
18	15.33	113.39
21	12.67	133.85
24	12.00	186.10
27	14.00	187.97
30	12.00	209.13

### 4.2. Comparison of SVM Performance with Data Reduction using K-SVNN

The authors conducted a comparison of prediction accuracy with and without reduction in SVM. The data with reduction will be processed by 2-Fold Cross-Validation. In SVM method, the authors use kernel: Linear, Radial Basis Function (RBF) and Polynomial. In this multi-class case, we use multi-class SVM approach [13, 22]. The authors use the Error Correction Output Code approach, using 3 digits binary code.

For performance time comparison, the authors compare total time spent SVM with and without reduction. For classification without reduction, the authors record the time spent during training and prediction. While the classification with data reduction, the authors record time spent during training and predictions, accompany with the time used K-SVNN to conduct data reduction.

The performance results for Linear kernel are presented in Table 2. From the table, it can be seen that the time used by SVM with data reduction is longer than without reduction. This is very reasonable because there are additional steps to do, data reduction. The longest time required reaches 360.85 ms and the shortest reaches 190.78 ms. While SVM without data reduction requires the longest time of 377.89 ms and the shortest 280.82 ms. The accuracy result given on the SVM with Linear kernel and with reduction has the highest accuracy 71.33%. While without reduction, the highest accuracy is 71.00%. As presented in Table 2.

Table 2. Time Comparison (Mili Second) and Reduction (Percentage) SVM with Linear Kernel

K	Time reduction with K-SVNN (ms) <sup>1</sup>	Training and prediction time SVM			Accuracy of SVM	
		Prediction <sup>2</sup>	With reduction Total time <sup>1+2</sup>	Without reduction	With reduction	Without reduction
3	50.09	140.69	<b>190.78</b>	<b>280.82</b>	63.00	67.00
6	51.83	182.43	234.26	294.46	64.33	67.00
9	58.39	204.29	262.68	<b>377.89</b>	60.67	67.00
12	68.90	221.91	290.81	289.61	67.33	68.67
15	66.28	243.42	309.7	310.32	66.33	68.00
18	71.32	247.77	319.09	290.08	62.67	65.67
21	81.16	253.50	334.66	294.56	67.33	70.33
24	77.43	251.41	328.84	297.58	<b>71.33</b>	<b>71.00</b>
27	85.53	262.48	348.01	308.66	68.33	68.67
30	95.59	265.26	<b>360.85</b>	309.79	62.33	64.33

The performance results for the RBF kernel are presented in Table 3. From the table, it can be seen that time spent of SVM with data reduction is also longer than without reduction, as in the results of Linear kernel. This is also due to additional steps to be taken, ie data reduction. The longest time required reaches 821.55 ms and the shortest time is 459.42 ms. While SVM without data reduction only takes the longest time 771.58 ms and the shortest time is 561.52 ms.

Accuracy results in SVM with RBF kernel with and without reduction got the same result, 33.33% on all K options used. As presented in Table 3. These results can be concluded that data reduction with multi-class K-SVNN can reduce data and preserve accuracy. The performance results for the Polynomial kernel are presented in Table 4. From the table, it can be seen that the time used SVM with and without reduction is long.

Table 3. Time Comparison (Mili Second), Reduction (Percentage), and Accuracy of SVM with RBF Kernel

K	Time reduction with K-SVNN (ms) <sup>1</sup>	Training and prediction time SVM			Accuracy of SVM	
		Prediction <sup>2</sup>	With reduction Total time <sup>1+2</sup>	Without reduction	With reduction	Without reduction
3	55.09	404.33	<b>459.42</b>	577.06	33.33	33.33
6	70.17	421.59	491.76	621.08	33.33	33.33
9	60.28	493.05	553.33	<b>561.52</b>	33.33	33.33
12	80.76	544.58	625.34	660.26	33.33	33.33
15	96.36	645.45	741.81	665.51	33.33	33.33
18	110.35	586.57	696.92	718.06	33.33	33.33
21	138.21	614.53	752.74	<b>771.58</b>	33.33	33.33
24	125.65	667.41	793.06	666.15	33.33	33.33
27	138.52	651.34	789.86	706.83	33.33	33.33
30	160.32	661.23	<b>821.55</b>	699.58	33.33	33.33

Table 4. Time Comparison (Mili Second), Reduction (Percentage), and Accuracy of SVM with Polynomial Kernel

K	Time reduction with K-SVNN (ms) <sup>1</sup>	Training and prediction time SVM			Accuracy of SVM	
		Prediction <sup>2</sup>	With reduction Total time <sup>1+2</sup>	Without reduction	With reduction	Without reduction
3	72.56	411.35	<b>483.91</b>	<b>475.94</b>	55.33	<b>54.67</b>
6	79.38	568.03	647.41	542.47	50.67	50.33
9	94.41	465.11	559.52	499.94	55.00	<b>54.67</b>
12	113.30	542.55	655.85	589.98	52.33	51.67
15	135.17	583.21	718.38	555.00	55.33	52.33
18	120.87	458.30	579.17	597.97	48.67	49.33
21	120.11	508.52	628.63	539.83	55.67	50.33
24	128.48	499.15	627.63	597.00	50.00	48.33
27	126.97	627.87	<b>754.84</b>	<b>607.68</b>	53.67	52.67
30	128.31	558.92	687.23	525.38	<b>57.00</b>	51.33

The longest time used by SVM with reduction is 754.84 ms at K=27 and the shortest time is 483.91 ms at K=3. While SVM without data reduction takes the longest time 754.84 ms and the shortest time is 475.94 ms. The accuracy obtained by SVM with Polynomial kernel without reduction have the highest accuracy 54.67% at K=3 and 9. While with data reduction, SVM obtained higher accuracy as in Linear kernels. The highest accuracy is 57.00% at K=30. As presented in Table 4.

#### 4.3. Comparison of Multi Class K-SVNN with other Data Reduction Method

The authors also perform performance comparisons with other data reduction methods namely CNN and TRKNN. The K parameter used by K-SVNN here is 5, relates to a reduction estimate similar to other methods. The alpha parameter used by TRKNN is 1.1. The authors compare prediction accuracy when data reduction results are used by SVM at classification sessions. In SVM classification sessions, the authors also compared the use of 3 SVM kernels: RBF, Linear, and Polynomial.

The results of the comparison are presented in Table 5. From the data presented in the table, it is known that the Multi-Class K-SVNN method obtains the highest reduction compared to other methods, the reduction rate is 72.67%. The highest accuracy is obtained by CNN on the Linear kernel, 66.33%, but for Polynomial kernel is obtained by K-SVNN, 60.33%. For RBF kernel, all methods get the same accuracy, 33.33%. It can be concluded that K-SVNN provides a better reduction rate and preserve the accuracy compared to other methods.

Table 5. The Comparison of Accuracy (%) with SVM using Data Reduction from Method

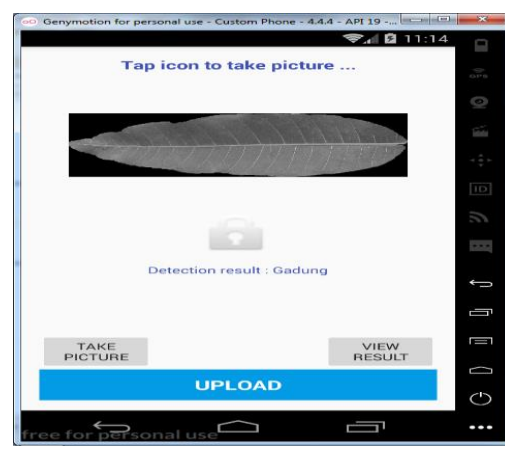
Method	Reduction (%)	Accuracy (%)		
		Linear	RBF	Polynomial
Multi-Class K-SVNN	<b>72.67</b>	62.33	33.33	<b>60.33</b>
CNN	72.00	<b>66.33</b>	33.33	54.33
TRKNN	70.67	61.67	33.33	51.33

#### 4.4. Implementation of Mango Leaf Classification on Android

The authors implement classification of mango leaf varieties in software that works on Android operating system. The authors use Android Studio 2.3.3 for system development. The testing is done on the emulator and phone with Android version 5 and 6. The software emulator used by the author is Genymotion version 2.9.0. The test environment used by the authors as follow: image acquisition on one leaf only, the time of image retrieval in the morning where the sun exposes mango leaves directly, the camera effect used is normal, the distance between the leaves and the camera about 10-20 cm. As shown in Figures 5. In Figure 5(a), it appears that the image acquisition can be done via camera or using files stored on the phone. While Figure 5(b) presents the results of the detection performed. As soon as detection process is complete, the 'see results' button can be used to display the detected image and the name of mango species obtained.



a. Capture the leaf



b. Result of detection

Figure 5. Application of mango leaf detection

## 6. Conclusion

From the experiment results, it can be concluded that multi-class K-SVNN as data training reduction method provides high reduction rate and preserve accuracy in mango leaf classification. The reduction rate achieved up to 62%, while the classification accuracy on data with and without reduction reached up to 71.33% and 71.00%, respectively. This study also proves that the accuracy obtained by SVM with Linear kernel is higher than other kernels. The performance comparisons with two other multi-class data reduction methods, i.e. CNN and TRKNN, showed that K-SVNN could produce better quality of data.

Suggestions for future research is related to time issue used for training process and predictions. The use of K-SVNN increases the duration of training due to the extra time required for reducing the data. A sophisticated handling method to shorten the training data process could be proposed.

## Acknowledgment

The authors give thanks to the Directorate of Research and Community Service (DRPM) DIKTI that fund Authors' research in the scheme of Inter-Universities Research Cooperation (PEKERTI) year 2017 between University of Bhayangkara Surabaya and Institut Teknologi Sepuluh Nopember, with contract number 72a/LPPM/V/2017 on 4 May 2017.

## References

- [1] S Agustin, Prasetyo E. *Klasifikasi Jenis Pohon Mangga Gadung Dan Curut Berdasarkan Tesktur Daun*. Seminar Nasional Sistem Informasi Indonesia. Surabaya. 2011.
- [2] E Prasetyo. Detection of Mango Tree Varieties Based on Image Processing. *Indonesian Journal of Science & Technology*. 2016; 1: 203-215.
- [3] E Prasetyo, Adityo RD, Suciati N, Fatchah C. Average and Maximum Weights in Weighted Rotation- and Scale-invariant LBP for Classification of Mango Leaves. *Journal of Theoretical and Applied Information Technology*. 2017.
- [4] RC Gonzalez, Wood RE. *Digital Image Processing*. Pearson Prentice Hall. 2008.
- [5] MB Bangun, Herdiyeni Y, Herliyana EN. Morphological Feature Extraction of Jabon's Leaf Seedling Pathogen using Microscopic Image. *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2016; 14: 254-261.
- [6] FY Manik, Herdiyeni Y, Herliyana EN. Leaf Morphological Feature Extraction of Digital Image Anthocephalus Cadamba. *TELKOMNIKA TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2016; 14: 630-637.
- [7] PE Hart. The condensed nearest neighbour rule. *IEEE Transactions on Information Theory*. 1968; 18: 515-516.
- [8] H Fayed, Atiya A. A novel template reduction approach for the k-nearest neighbor method. *IEEE Transactions on Neural Networks*. 2009; 20: 890-896.
- [9] E Prasetyo. K-Support Vector Nearest Neighbor Untuk Klasifikasi Berbasis K-NN. in *Seminar Nasional Sistem Informasi Indoensia*, Surabaya. 2012: 245-250.
- [10] E Prasetyo, Rahajoe RAD, Agustin S, Arizal A. Uji Kinerja dan Analisis K-Support Vector Nearest Neighbor Terhadap Decision Tree dan Naive Bayes. *Eksplora Informatika*. 2013; 3: 1-6.
- [11] E Prasetyo, Alim S, Rosyid H. *Uji Kinerja dan Analisis K-Support Vector Nearest Neighbor Dengan SVM Dan ANN Back-Propagation*. Seminar Nasional Teknologi Informasi dan Aplikasinya. Malang. 2014.
- [12] E Prasetyo. K-Support Vector Nearest Neighbor: Classification Method, Data Reduction, and Performance Comparison. *Journal of Electrical Engineering and Computer Sciences*. 2016; 1: 1-6.
- [13] P Tan, Steinbach M, Kumar V. *Introduction to Data Mining*. Boston San Fransisco New York: Pearson Education. 2006.
- [14] AB Mutiara, Refianti R, Mukarromah NRA. Musical Genre Classification Using Support Vector Machines and Audio Features. *TELKOMNIKA TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2016; 14: 1024-1034.
- [15] M Hamiane, Saeed F. SVM Classification of MRI Brain Images for Computer-Assisted Diagnosis. *International Journal of Electrical and Computer Engineering (IJECE)*. 2017; 7: 2555-2564.
- [16] S Theodoridis, Koutroumbas K. *Pattern Recognition*. Burlington, MA, USA: Academic Press. 2009.
- [17] YA Sari, Dewi RK, Fatchah C. Seleksi Fitur Menggunakan Ekstraksi Fitur Bentuk, Warna, Dan Tekstur Dalam Sistem Temu Kembali Citra Daun. *JUTI*. 2014; 12: 1-8.

- [18] E Prasetyo. Reduksi Data Latih Dengan K-SVNN Sebagai Pemrosesan Awal Pada ANN Back-Propagation Untuk Pengurangan Waktu Pelatihan. *SIMETRIS*. 2015; 6: 223-230.
- [19] I Tomek. Two modifications of cnn. *IEEE Transactions on systems, Man, and Cybernetics*. 1976; 6: 769-72.
- [20] E Prasetyo, Adityo RD, Suciati N, Fatchah C. Deteksi Wilayah Cahaya Intensitas Tinggi Citra Daun Mangga Untuk Ekstraksi Fitur Warna dan Tekstur Pada Klasifikasi Jenis Pohon Mangga. Presented at the Seminar Nasional Teknologi Informasi, Komunikasi dan Industri, UIN Sultan Syarif Kasim Riau. 2017.
- [21] E Prasetyo, Adityo RD, Suciati N, Fatchah C. *Mango Leaf Image Segmentation on HSV and YCbCr Color Spaces Using Otsu Thresholding*. *International Conference on Science and Technology*, Yogyakarta. 2017.
- [22] E Prasetyo. *Data Mining-Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: Andi Offset. 2014.